

Universidad Carlos III de Madrid
Escuela Politécnica Superior



Grado en Ingeniería Telemática
Trabajo Fin de Grado

Herramienta de pruebas funcionales en la nube de servicios RCS

Autor: Ricardo Martínez Barrero

Tutor: Jesús Arias Fisteus

Director: Miguel Ángel Núñez (Mántica solutions)

Agradecimientos

Para el desarrollo profesional o personal de cualquier individuo es necesario tener la oportunidad enfrentarse a nuevos retos. Las situaciones nuevas o desconocidas provocan que las personas tengan que tomar decisiones propias y para ello comience una fase de análisis y elección de la mejor solución. Este proceso interno ayuda a ser más consciente y a adelantarse a posibles eventualidades en casos futuros.

Quiero agradecer a Mántica solutions la oportunidad que me han dado al trabajar con ellos en esta etapa de nuevos proyectos y grandes retos. Gracias por los nuevos retos, gracias por hacerme crecer y mejorar cada día.

Por otro lado, sin el apoyo incondicional de mi familia no hubiera sido posible llegar hasta este momento, siempre os necesito conmigo y me siento muy orgulloso de forma parte de vosotros.

Por último quiero agradecer a todo aquel que de alguna manera ha influido en mi, a través de la formación, de la amistad o simplemente con un comportamiento o forma de ser que haya servido de ejemplo, y que espero pueda llegar a emular algún día.

Resumen

Este proyecto surge como una oportunidad de negocio debido al incremento de la demanda de servicios profesionales en el ámbito de pruebas extremo a extremo de servicios RCS durante los últimos meses en la compañía.

En la mayoría de los casos los proveedores de servicios RCS no cuentan con las herramientas adecuadas para diagnosticar el estado de sus redes en tiempo real. Se utilizan procedimientos manuales que suelen ser imprecisos e insuficientes. Esto supone un riesgo en la calidad del servicio ofrecido a los usuarios y además un sobrecoste.

Para intentar dar soluciones a dichos clientes se pretende desarrollar una herramienta automatizada de monitorización que sea capaz de diagnosticar la disponibilidad de varios servicios RCS de forma rápida y sencilla.

Además se pretende que la integración, ampliación y soporte sea flexible y fácil de gestionar.

Índice

| | | |
|----------|--|-----------|
| 1 | Introducción | 8 |
| 1.1 | Introducción | 8 |
| 1.2 | Entorno..... | 8 |
| 1.3 | Motivación..... | 8 |
| 1.4 | Objetivos | 9 |
| 1.5 | Contenido de la memoria | 9 |
| 2 | Estado del arte | 12 |
| 2.1 | Introducción | 12 |
| 2.2 | Herramientas de monitorización..... | 12 |
| 2.3 | Análisis de la dinámica actual de pruebas..... | 13 |
| 2.4 | Conclusiones..... | 14 |
| 3 | Claves del proyecto | 16 |
| 3.1. | Introducción | 16 |
| 3.2. | Planteamiento | 16 |
| 3.3. | Clientes | 16 |
| 4 | Implementación..... | 19 |
| 4.1 | Introducción | 19 |
| 4.2 | Requerimientos | 19 |
| 4.3 | Funcionalidad | 19 |
| 4.3.1 | <i>Prerrequisitos</i> | <i>19</i> |
| 4.3.2 | <i>Monitorización de servicios RCS</i> | <i>20</i> |
| 4.3.3 | <i>Interfaz</i> | <i>26</i> |
| 4.3.4 | <i>Documentación</i> | <i>30</i> |
| 5 | Despliegue y certificación..... | 33 |
| 5.1 | Despliegue | 33 |
| 5.1.1 | <i>Servidor web.....</i> | <i>33</i> |
| 5.1.2 | <i>Conectividad.....</i> | <i>34</i> |
| 5.1.3 | <i>Usuarios reales de operadores que soporten el servicio.....</i> | <i>34</i> |
| 5.1.4 | <i>Sistema de seguridad</i> | <i>34</i> |
| 5.2 | Pruebas..... | 34 |
| 6 | Planificación y presupuesto..... | 37 |
| 6.1 | Planificación | 37 |
| 6.2 | Presupuesto..... | 37 |
| 6.2.1 | <i>Costes de hardware</i> | <i>38</i> |
| 6.2.2 | <i>Coste de software.....</i> | <i>38</i> |
| 6.2.3 | <i>Costes de personal.....</i> | <i>38</i> |
| 6.2.4 | <i>Totalización de costes.....</i> | <i>39</i> |
| 7 | Futuro y Conclusiones | 41 |
| 7.1 | Futuro | 41 |
| 7.1.1 | <i>Mejoras hardware</i> | <i>41</i> |
| 7.1.2 | <i>Mejoras software</i> | <i>41</i> |
| 7.1.3 | <i>Nuevas líneas de negocio</i> | <i>42</i> |
| 7.2 | Conclusiones..... | 42 |
| 8 | Bibliografía | 44 |

1 Introducción

1.1 Introducción

Todo sistema que evoluciona está condicionado a tener fallos. Esta es una de las bases fundamentales de las pruebas regresivas. Debido a los cambios suelen aparecer problemas de integración, actualización, compatibilidad y normativa.

En entornos evolutivos es difícil obtener un diagnóstico actualizado del estado si no se poseen las herramientas adecuadas, y en muchos casos el trabajo está demasiado enfocado a avanzar el desarrollo y no tanto en la calidad del mismo.

Las redes RCS(*Rich Communication Services*)[1] están en desarrollo en muchos operadores de telefonía en todo el mundo y han sufrido variaciones para dar soporte a nuevas funcionalidades y corregir defectos. Además al tratarse de una tecnología reciente, a penas dos años y medio desde su lanzamiento, y específica es difícil encontrar expertos en la misma.

Este escenario resulta muy apropiado para buscar soluciones a las necesidades que se han mencionado.

1.2 Entorno

El proyecto se ha realizado en el ámbito de beca de colaboración en empresa, incluyendo la formación teórica necesaria en RCS para realizar la parte práctica en la que se basa el proyecto.

Mántica es una empresa dedicada a ofrecer servicios profesionales de certificación y test de productos de telecomunicación en diferentes capas de protocolo. Desarrolla su actividad profesional con empresas de todo el mundo, principalmente operadores de telefonía y empresas suministradoras.

El departamento de servicios profesionales de la compañía, realiza pruebas “*End to End*” de certificación de terminales, equipos y redes RCS, con el fin de acreditarlos.

1.3 Motivación

El comienzo de la beca de colaboración coincidió con la finalización de un proyecto propio de la empresa, que provee los prerequisites necesarios para este.

La compañía desarrolló una librería RCS que permitía abstraer al desarrollador de las pruebas de la mensajería a bajo nivel, para optimizar la realización de los escenarios de prueba y separar la dependencia del nivel de protocolo. De esta forma la evolución de la librería en relación con el estándar y de las pruebas en relación con la demanda de los clientes queda perfectamente diferenciada.

Por otra parte, la experiencia de varios años realizando pruebas en entornos similares y la apuesta de futuro que la compañía deposita en la tecnología RCS, propicia este proyecto interno en el ámbito del I+D.

En último lugar, y tras analizar las soluciones que se ofrecen en este campo en el mercado, se observa una oportunidad de ampliar los servicios y sistemas que la compañía ofrece a sus clientes.

1.4 Objetivos

El principal objetivo del proyecto es crear una herramienta de monitorización activa de redes RCS que cumpla con las siguientes características:

- Detectar indisponibilidades de servicios RCS en tiempo real o bajo demanda.
- Automatizar un subconjunto de pruebas frecuentes en las tareas de test diarias de los ingenieros de la empresa.
- Crear una herramienta accesible y multiplataforma para todo tipo de clientes.
- Crear una herramienta flexible y fácil de evolucionar.
- Encontrar un modelo de negocio en línea con las especificaciones.

1.5 Contenido de la memoria

Con esta memoria se pretende detallar el proceso de desarrollo de una herramienta de monitorización activa de redes RCS. El documento se ha estructurado en capítulos para hacer más sencillo su seguimiento.

El primer capítulo está compuesto por la introducción, el entorno en el que se gesta el proyecto, las motivaciones que llevan a realizarlo y los objetivos iniciales que se marcaron.

En el segundo capítulo, el estado del arte, se explican los procedimientos y tecnologías que se utilizaban para realizar este tipo de pruebas en la compañía y en algunos clientes antes de desarrollar esta herramienta. Mediante el proceso de análisis y revisión podemos extraer muchos requisitos y futuras ideas para incorporar en la herramienta antes incluso de diseñarla.

Más adelante, en el tercer capítulo, claves del proyecto, repasaremos cuales han sido los eventos más relevantes que han hecho que se llevase a cabo, y algunas experiencias personales que varían el diseño inicial del proyecto.

El cuarto capítulo está dedicado al desarrollo del proyecto, donde se detalla técnicamente la herramienta de monitorización y se explican todas las partes de la misma, así como una justificación del entorno y requisitos.

Posteriormente, en el capítulo quinto, se expone el diagrama de despliegue detallando los elementos que lo forman y se analiza la fase de pruebas realizadas.

El capítulo sexto está dedicado a la planificación del proyecto y estimaciones previstas en recursos personales y materiales además del presupuesto que se ha destinado a la realización de la herramienta.

Para finalizar, se analizan las líneas futuras de desarrollo, mejoras y proyectos asociados y se exponen las conclusiones extraídas del proyecto en el capítulo siete.

2 Estado del arte

2.1 Introducción

El objetivo de este capítulo es analizar los métodos y herramientas previos a la realización del proyecto, obteniendo justificación para la realización del mismo y ventajas e inconvenientes de sustituir los anteriores por la nueva herramienta.

2.2 Herramientas de monitorización

La monitorización de una red consiste en la comprobación constante de funcionamiento y actividad de los equipos que la componen para detectar anomalías y notificarlas.

Todos los equipos implementan monitores de actividad, servicio o fallo, lo cual permite analizar cualquier anomalía que hayan tenido. El mayor problema es que se trata de un post análisis, ya que en el supuesto caso de que equipo falle el monitor embebido fallará también.

Existen también subsistemas de red para administradores, que permiten desde un equipo central monitorizar un grupo de equipos. La ventaja frente a los monitores embebidos es que la monitorización es externa y por lo tanto un fallo en el equipo no limita el seguimiento. Con este tipo de herramientas podríamos solucionar una incidencia en tiempo real.

Como ejemplo de estos sistemas podemos nombrar Nagios[2] o Aggregate[3] con funcionalidades avanzadas como la creación autónoma de topologías de red, escalabilidad en equipos y crecimiento bajo demanda funcionalidades. Además son de gran utilidad para protocolos estandarizados, pero básicamente estudian el estado de la red basándose en la conectividad y la actividad.

Las redes IMS (*IP Multimedia Core Network Subsystem*) [4]/RCS están compuestas por numerosos equipos con funcionalidades y tareas diferentes. Esto supone que los puntos de fallo se dividen por toda la arquitectura y son más difíciles de encontrar y diagnosticar.

La monitorización de servicios requiere una implementación de los mismos, ya que puede no haber fallos a nivel de conectividad o actividad, pero alguna funcionalidad no está activa. Este tipo de monitorización es activa, ya que no somos meros observadores de la red, sino que interactuamos con ella creando tráfico nuevo.

Otro punto bloqueante en redes de operadores es el acceso a la infraestructura de la red. Los sistemas que hemos mencionado monitorizan la red estando conectados a ella directamente, para tener visibilidad de todos los equipos que la componen. En estos casos los clientes van a acceder a los servicios desde fuera de las redes, y puede

que un fallo de acceso que deja sin servicio a todos los clientes no sea reconocido por una herramienta de monitorización que se encuentra en la red.

Por último, al tratarse de entornos productivos, con usuarios reales, es importante que las herramientas de monitorización sean lo más parecidas a la realidad posible. Los casos de estudio a diseñar deben simular los comportamientos habituales de los usuarios. Para este servicio no existe ninguna herramienta con dichas características.

2.3 Análisis de la dinámica actual de pruebas

El servicio RCS, en su denominación comercial Joyn®[5] surgió, para muchos operadores de telefonía, como una alternativa a los servicios de mensajería instantánea, que se ofrecen a través de aplicaciones propietarias que desarrollan sus propios protocolos y sistemas y ofrecen el servicio a través de internet.

Joyn® es para muchos la oportunidad de estandarizar la mensajería instantánea, enriqueciendo el servicio de SMS/MMS que ha visto descender su cuota de utilización durante los últimos años, y añadiendo nuevas funcionalidades muy demandadas y que aportan valor añadido a dicho servicio. Como el intercambio de mensajes y ficheros, vídeo y audio en tiempo real y soporte para varios dispositivos.

Las posibilidades que ofrece RCS, no pasan sólo por ser una aplicación de mensajería instantánea más. Podemos pensar en escenarios en los que un servicio estandarizado con dichas características resultaría de gran utilidad. Como ejemplo, un videojuego en red, podría proveer al usuario la funcionalidad de enviar y recibir mensajes en un chat con los demás jugadores, además estos mismos tendrían la posibilidad de compartir herramientas necesarias para el juego e incluso realizar modos de campaña con otros usuarios con audio y video en tiempo real.

Otras ventajas importantes para los operadores son el valor añadido para sus redes y el ahorro en recursos.

El modelo de aplicaciones para terminales, hace que los operadores sean relegados a simples intermediarios que ofrecen conectividad a servicios de terceros a los usuarios. Con este modelo el servicio permanece en la red, añadiendo valor a la misma en la comparativa con otros operadores que no poseen servicios similares, abaratando costes de transporte e interconexión, ya que no es necesario rutar el tráfico fuera de la red y permitiendo nuevos modelos de negocio. Por ejemplo, la utilización del servicio sin cortes, ni límites de velocidad de conexión una vez que el usuario haya finalizado su cuota de conexión del periodo vigente.

Ya que el servicio se ofrece como ampliación del SMS/MMS tradicional, los operadores pretender incluirlo como aplicación nativa o propia en los terminales de sus usuarios.

Debido a lo anterior, el terminal se vuelve una herramienta básica para realizar las pruebas para los ingenieros que prueban de este servicio y es la herramienta de monitorización RCS más usada.

2.4 Conclusiones

A primera vista puede resultar fácil y rápido realizar los test con una aplicación móvil preparada para usar el servicio, pero la experiencia demuestra que este tipo de herramienta tiene ciertas carencias y limitaciones:

- El comportamiento de la aplicación no es siempre el esperado.

Las aplicaciones que dan cobertura a este servicio también deben ser probadas y certificadas. En las situaciones en las que las normas dejan parte del diseño a elección del desarrollador, puede aparecer escenarios que si bien, no son imprescindibles para la certificación de un cliente, si lo son para la red. Dichos escenarios no podrían ser probados con tan sólo esta herramienta.

- No es posible realizar la misma prueba dos veces.

Debido a que la prueba se realiza de forma manual, es difícil en ocasiones reproducir escenarios erróneos.

- Las aplicaciones están diseñadas para cumplir con las expectativas de utilización de los usuarios finales.

En muchas ocasiones enmascaran comportamientos erróneos de red y confunden a los propios ingenieros de pruebas.

- No se pueden realizar todas las pruebas requeridas para la acreditación.

Existen pruebas que pretenden determinar el comportamiento en escenarios con fallos que son imposibles de simular con dichas aplicaciones.

- El proceso de pruebas es largo.

La realización manual de las pruebas requiere tiempo para estudiar la casuística, preparar el escenario, obtener toda la información disponible para analizarla, llegar a un diagnóstico.

Debido a todos estos inconvenientes parece más que justificado desarrollar una herramienta de monitorización RCS que solucione todos estos problemas.

3 Claves del proyecto

3.1. Introducción

Debido al descenso en la demanda de servicios en algunos departamentos de la empresa, se decide, tras estudiar todas las opciones, continuar con la evolución de la librería RCS y crear servicios destinados a dar soporte a clientes de forma inmediata e independiente.

Crear un servicio web bajo demanda que permita a empresas disponer de pruebas de monitorización activa de servicios en cualquier momento y lugar, de forma automática, fiable y continua es un objetivo difícil de conseguir si no se estructura bien el trabajo.

3.2. Planteamiento

El proyecto surge internamente para probar algunos componentes que la empresa había desarrollado y para familiarizar al alumno con la tecnología.

Aprovechando los conocimientos que los miembros del equipo poseen de la tecnología, y la experiencia en el campo de la realización de pruebas, surge la idea de aplicarlos y crear una herramienta única que realice el trabajo mecánico de ejecutar una prueba, pero que también sea capaz de diagnosticar problemas de fondo o forma con el protocolo y la funcionalidad de la red.

De esta manera reduciríamos el tiempo que hace falta para ejecutar y encontrar anomalías en las redes, lo que supone un ahorro en recursos y tiempo.

3.3. Clientes

Uno de los puntos más críticos del proyecto se produjo cuando quedaban 2 semanas para la finalización del mismo.

Un cliente, al que se le había informado del servicio, pide una prueba de concepto para ver como funciona la herramienta de verdad y que ventajas puede proporcionarle a él particularmente.

Por nuestra parte todavía no habíamos terminado la certificación del entorno definitivo pero necesitábamos que la impresión del cliente fuese que teníamos un sistema funcional y listo para su comercialización inmediata.

Esas dos semanas fueron largas, combinando reuniones para determinar las prioridades y cumplir los plazos con un desarrollo consciente y riguroso. Cuantos menos fallos cometiese, más corto sería el proceso de depuración y pruebas.

Al final todo salió bien. El cliente quedó muy satisfecho con la prueba ofrecida y utilizó el sistema por su cuenta durante las siguientes semanas sin problemas ni incidencias, aunque, para mi tranquilidad, yo también realizaba algunas pruebas y revisaba los archivos con la información de utilización que se van almacenando en el servidor.

4 Implementación

4.1 Introducción

En este capítulo se expone y analiza la herramienta realizada, detallando cada una de sus partes.

4.2 Requerimientos

Tras analizar los capítulos anteriores se llegó a una lista de requisitos que debía cumplir la herramienta:

Portable y escalable. Debía poder utilizarse en cualquier lugar y debe ser fácil crecer en clientes y funcionalidad.

Se decidió que el servicio se ofrecerá a través de internet, los usuarios se conectarían a la plataforma de pruebas de manera independiente y remota. Además el diseño cliente-servidor permite poder ampliar la capacidad y la funcionalidad del entorno bajo demanda sin que afecte en nada a los clientes.

4.3 Funcionalidad

Como ya se ha dicho la herramienta ha sido creada para proveer a los clientes de la empresa de un sistema automatizado de monitorización activa de algunos de los servicios básicos RCS más importantes.

Para poder realizar dicha funcionalidad debemos tener en cuenta los requisitos previos que afectan a algunas decisiones estructurales del diseño.

4.3.1 Prerrequisitos

4.3.1.1 Protocolo RCS

En primer lugar el desarrollador del proyecto debe tener conocimientos sobre el servicio RCS y su protocolo. Para ello una fase previa a la implementación de la herramienta, es el estudio del protocolo.

4.3.1.2 Librería RCS

El módulo más importante que sirve de punto de partida para la realización del proyecto es la librería RCS que la compañía ha desarrollado previamente.

Para la realización del proyecto se necesitó una copia de la librería, disponible únicamente para sistemas operativos Microsoft Windows[6], una licencia, y la documentación de instalación y uso.

4.3.1.3 Herramientas de desarrollo

Al tratarse de un proyecto de carácter práctico, se requiere la utilización de herramientas de desarrollo.

Ya que la herramienta es un servicio web, que tiene que utilizar el sistema operativo Windows, y necesitamos una base de datos para la gestión de clientes y accesos, un lenguaje de programación versátil, compatible con entornos web y que conozca el desarrollador y un servidor web que lo aloje, se decidió usar Wampserver 2.2[7] como entorno de desarrollo. Esta herramienta está compuesta por:

- Php MyAdmin 3.5.1
- PHP 5.4.3
- MySQL 5.5.24
- Apache 2.4.2

Además la interfaz se ha realizado en:

- HTML5[8]
- CSS3[9]
- JavaScript[10]
- JQuery 1.9.1[11]

4.3.2 Monitorización de servicios RCS

Para la monitorización de la red se decidió implementar escenarios que un usuario real del sistema implementase con frecuencia.

4.3.2.1 Casos de prueba

Se definieron varios escenarios específicos de pruebas para ser implementados.

Todos los escenarios cumplen la premisa de que los usuarios son RCS y pueden acceder a la red.

Por lo tanto los usuarios que se utilizaran para las pruebas han sido previamente dados de alta en el sistema y se han obtenido sus parámetros específicos de configuración de la red.

4.3.2.1.1 Intercambio de capacidades

Para saber si un contacto del usuario A es RCS, es necesario intercambiar capacidades con el. De esta manera el usuario B muestra al usuario A las capacidades que implementa y viceversa.

Dos usuario pueden intercambiar un fichero, si ambos tienen la capacidad de hacerlo y lo han anunciado a través del intercambio de capacidades.

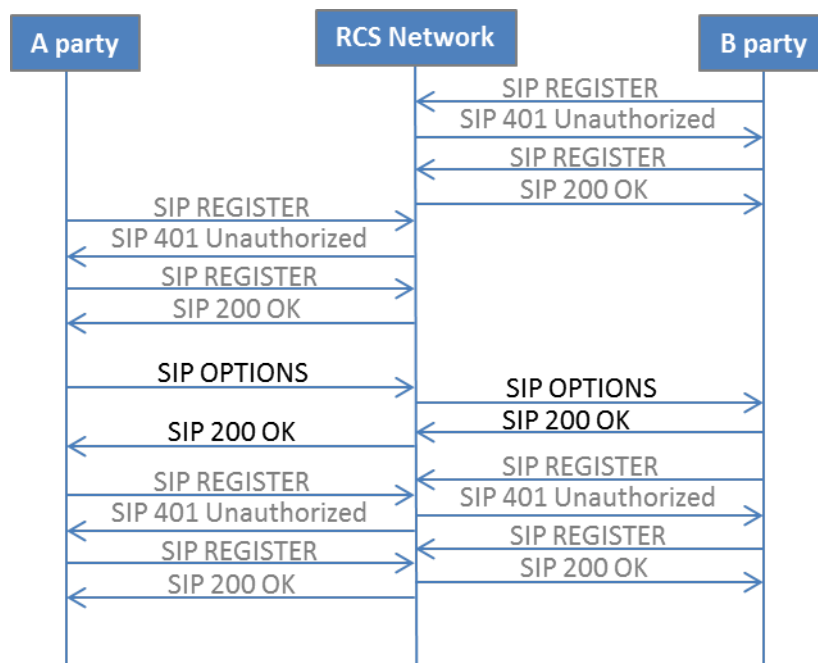


Figura 4.3.2.1 Diagrama de flujo del intercambio de capacidades

4.3.2.1.2 Chat 1 a 1

El usuario A envía un mensaje al usuario B, que a su vez envía las notificaciones de entrega y visualización del mensaje.

Para que este caso se parezca lo máximo posible a un chat real, se simula que el usuario B tarda un poco en abrir la ventana de chat, lo que provocaría el establecimiento de la sesión MSRP[12] mediante el mensaje definitivo 200 OK a la transacción SIP.

El usuario B manda el mensaje de entrega a través de SIP ya que todavía no ha establecido la sesión. Una vez que la sesión MSRP se establece el resto de mensajes se deben enviar en dicha sesión. El mensaje de visualización se envía dentro de la sesión.

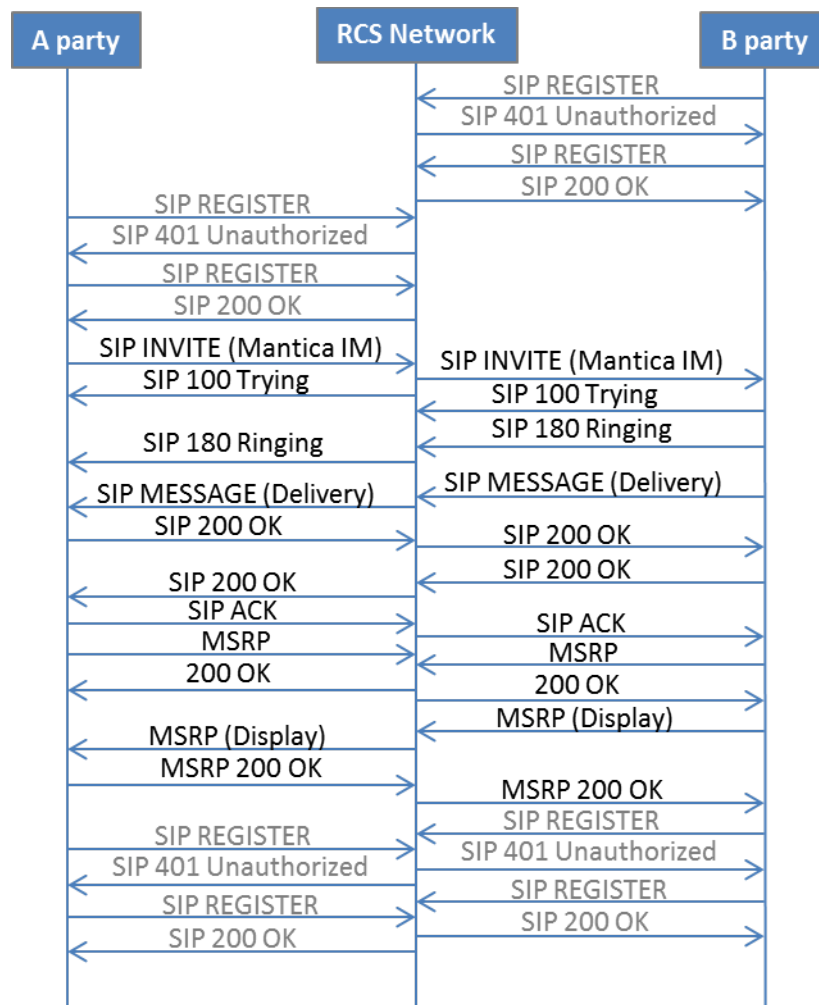


Figura 4.3.2.2 Diagrama de flujo del chat 1 a 1

4.3.2.1.3 Envío de ficheros sobre MSRP

En este caso el usuario A manda un fichero al usuario B, suponiendo que este no tiene la capacidad de utilizar la transferencia de ficheros a través sobre HTTP y así lo ha anunciado en sus capacidades.

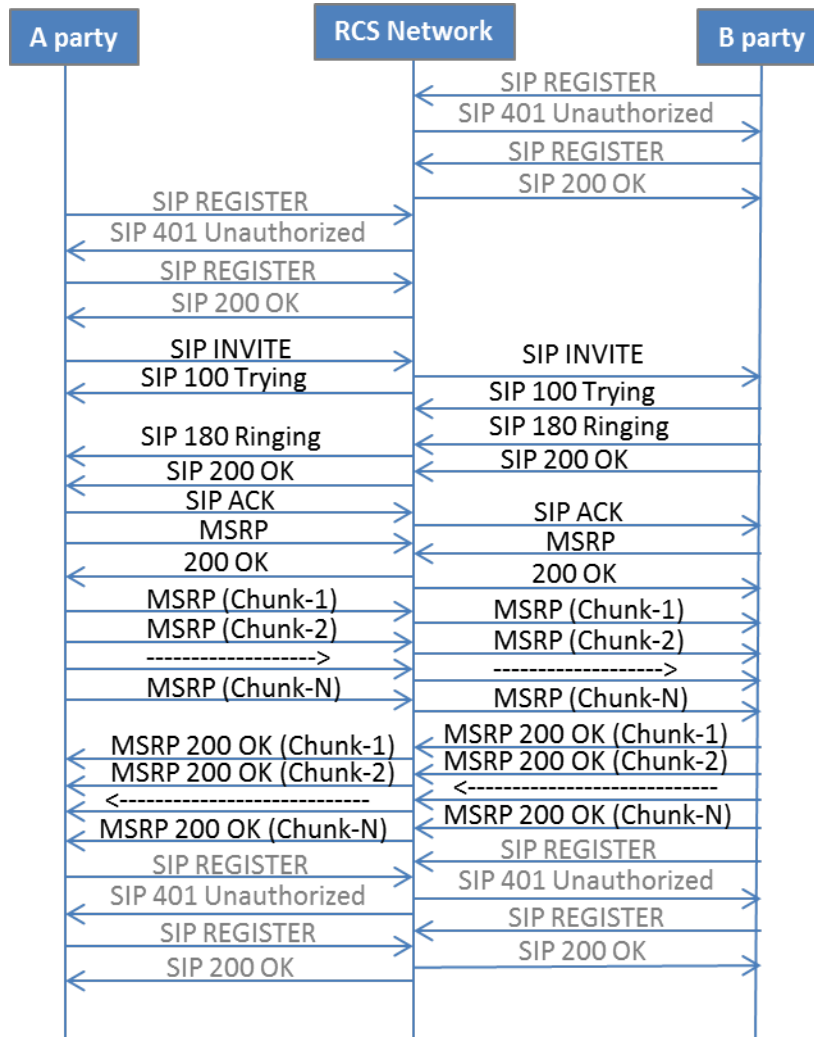


Figura 4.3.2.3 Diagrama de flujo del envío de fichero sobre MSRP

4.3.2.1.4 Envío de ficheros sobre http

El usuario A envía un fichero al usuario B, para ello el usuario A debe subir el fichero a través del protocolo HTTP al contenedor de archivos, y enviar la dirección donde se ha almacenado el archivo para que el usuario B pueda descargarlo y visualizarlo. El usuario B envía notificaciones de entrega y visualización.

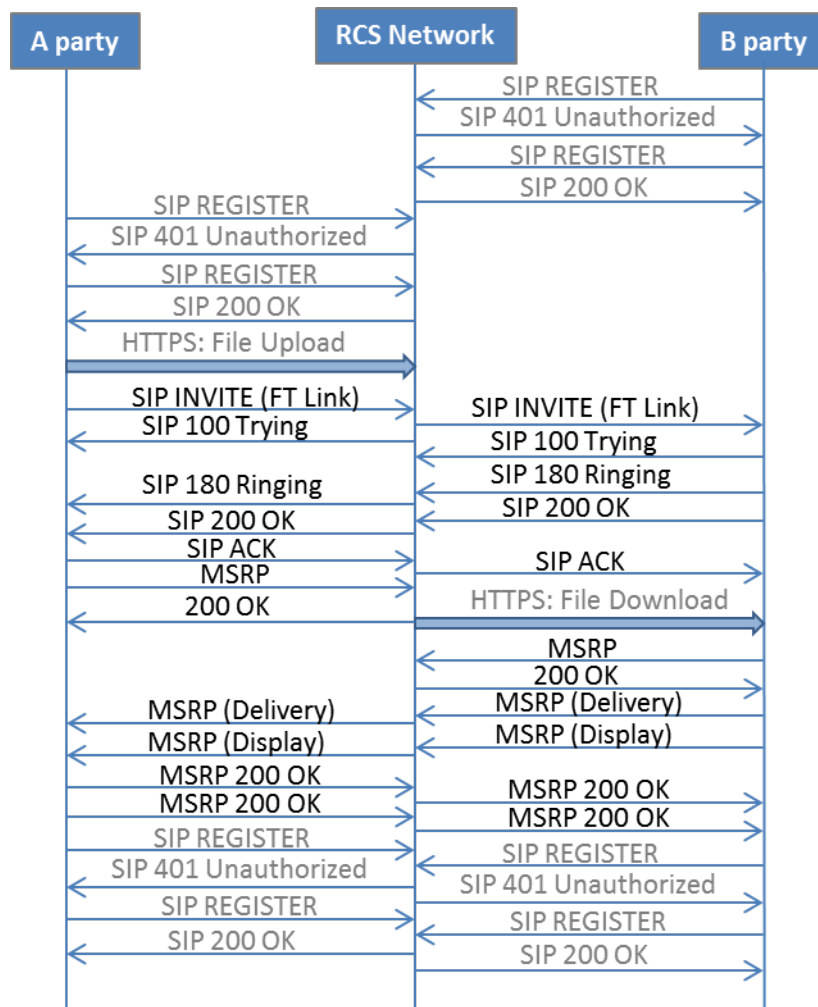


Figura 4.3.2.4 Diagrama de flujo del envío de ficheros sobre HTTP

4.3.2.1.5 Envío de vídeo.

El usuario A comparte vídeo en tiempo real de su cámara con el usuario B a través del protocolo RTP[13].

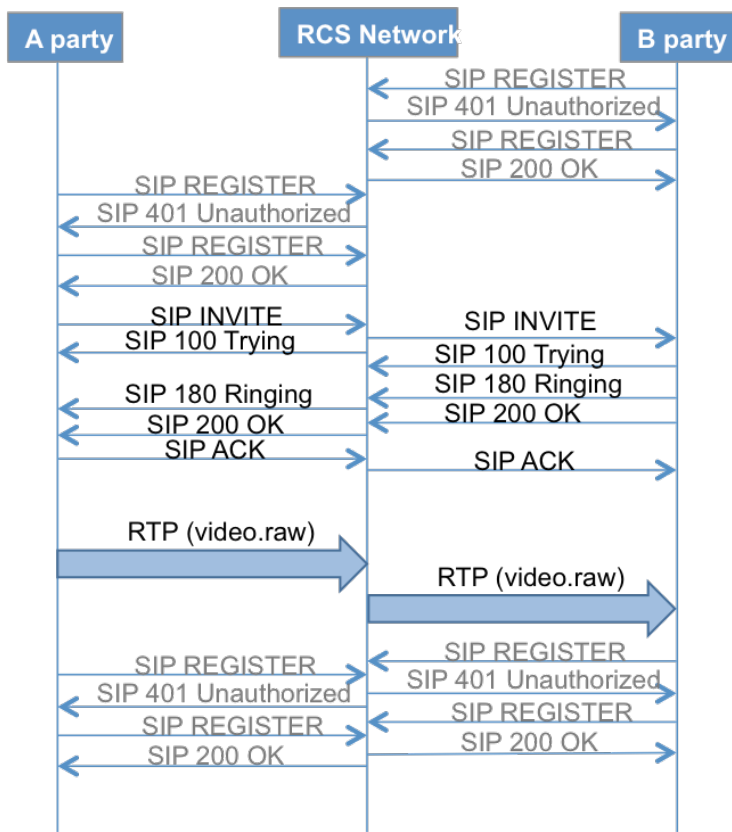


Figura 4.3.2.5 Diagrama de flujo del envío de vídeo

4.3.2.2 Ampliación de casos de prueba

Para completar la funcionalidad de la herramienta se ha definido un lenguaje para crear escenarios sencillos, que permite subir un archivo de texto con los pasos necesarios y ordenados para ejecutar un escenario propio.

El lenguaje consta de algunas funciones, métodos, a lo que se le añaden parámetros para crear una lista de ejecución de un escenario

El formato de archivo de entrada es el siguiente:

```
Methods
CREATE
DELETE
REGISTER
DE-REGISTER
CAPABILITIES
1T01-CHAT
GROUP-CHAT
FILE-TRANSFER

Line Format:

CREATE MSISDN User_ID
DELETE User_ID
REGISTER User_ID
DE-REGISTER User_ID
CAPABILITIES From_User_ID To_User_ID
1T01-CHAT From_User_ID To_User_ID IM/FT/DELETE "TEXT/FILE"

                                (IM) "IM"
                                (FT) "/file"
                                (DELETE)

GROUP-CHAT GC_ID From_User_ID CREATE/IM/FT/DELETE PARAMETERS

                                (CREATE) To_User_ID1 To_User_ID2 To_User_ID3
                                (IM) "IM"
                                (FT) "/file"
                                (DELETE)

Examples:

CREATE +346666666666 UserA
REGISTER UserA

CREATE +347777777777 UserB
REGISTER UserB

CAPABILITIES UserA UserB

1T01-CHAT UserA UserB IM "This is UserA"
1T01-CHAT UserA UserB FT "/images/UserA_car.jpg"
1T01-CHAT UserB UserA IM "This is UserB"
1T01-CHAT UserB UserA FT "/images/UserB_car.jpg"

CREATE +348888888888 UserC
REGISTER UserC

CAPABILITIES UserC UserA
CAPABILITIES UserC UserB

GROUP-CHAT "TEST_GC" UserC CREATE UserA UserB
GROUP-CHAT "TEST_GC" UserC IM "Hello, this is C"
GROUP-CHAT "TEST_GC" UserB IM "Hello, this is B"
GROUP-CHAT "TEST_GC" UserA IM "Hello, this is A"
GROUP-CHAT "TEST_GC" UserB FT "/images/group_photo.jpg"
GROUP-CHAT "TEST_GC" UserA DELETE

DE-REGISTER UserA
DELELE UserA
DE-REGISTER UserB
DELELE UserB
DE-REGISTER UserC
DELELE UserC
```

Figura 4.3.2.6 Formato de entrada de escenarios de prueba propios

4.3.2.3 Resultados de la monitorización

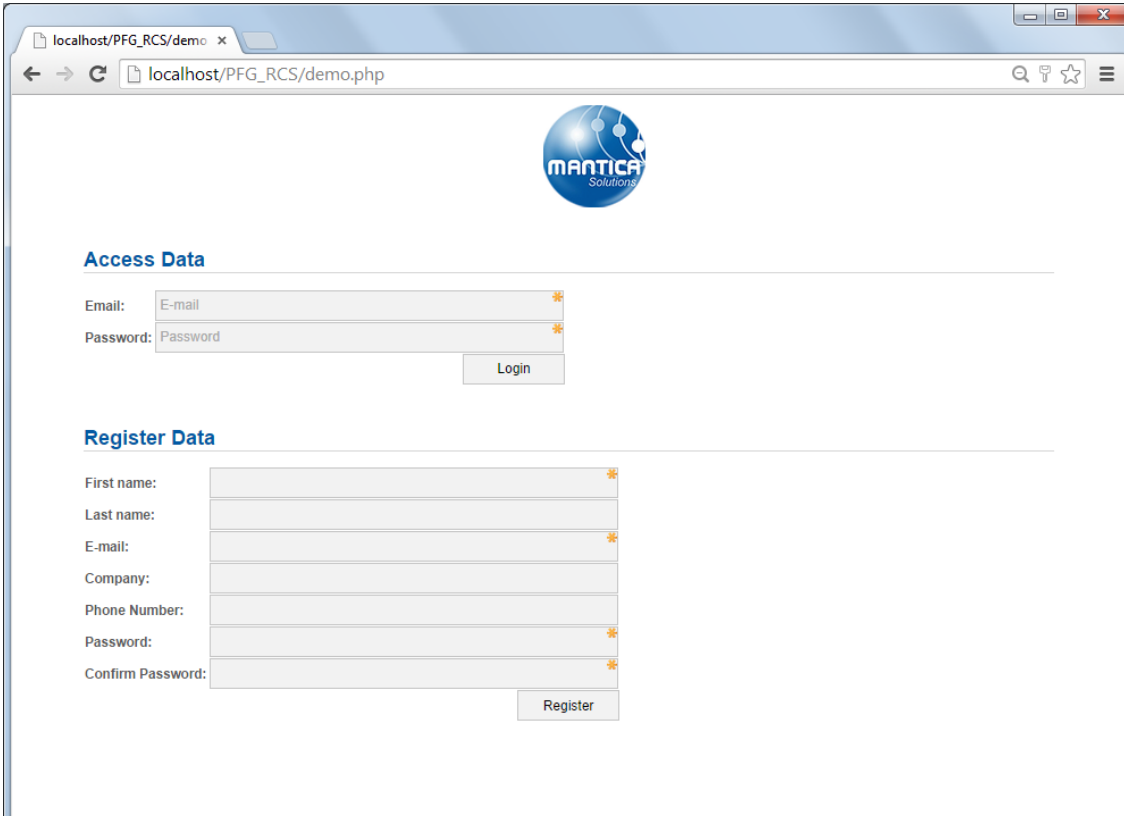
Tanto los escenarios predefinidos como los escenario propios devuelven información a través de una tabla de registro al cliente. Además se incluyen KPIs (Key Performance Indicator) [14] de tiempos de ejecución para las transacciones principales.

4.3.3 Interfaz

Se opta por una solución web, accesible desde cualquier navegador, que sea fácil de entender y manejar por cualquier usuario.

4.3.3.1 Identificación de usuarios y control de acceso

Debido a que la herramienta está accesible desde internet es necesario implementar un control de acceso y gestión de usuarios a la plataforma. Para ello se crea una base de datos con información relevante para la identificación y el control de los mismos en el sistema.



The screenshot shows a web browser window with the address bar displaying 'localhost/PFG_RCS/demo.php'. The page features the MANTICA Solutions logo at the top center. Below the logo, there are two main sections: 'Access Data' and 'Register Data'. The 'Access Data' section contains input fields for 'Email' and 'Password', each with a small orange asterisk icon to its right, and a 'Login' button. The 'Register Data' section contains input fields for 'First name', 'Last name', 'E-mail', 'Company', 'Phone Number', 'Password', and 'Confirm Password', each with a small orange asterisk icon to its right, and a 'Register' button.

Figura 4.3.3.1 Formulario de acceso a la herramienta

4.3.3.2 Formulario de selección de pruebas

Para la elección de las pruebas también se crea un interfaz con un formulario donde el usuario tiene menús desplegables con la información necesaria para seleccionar las pruebas.

localhost/PFG_RCS/demo x

localhost/PFG_RCS/demo.php

joyn™ users

A-Party Operator A: +34111111111

B-Party Operator A: +34999999999

Test Cases

| | |
|-------------------------|-------------------------------------|
| Register | <input checked="" type="checkbox"/> |
| Capabilities | <input checked="" type="checkbox"/> |
| 1 to 1 chat | <input type="checkbox"/> |
| File transfer over MSRP | <input type="checkbox"/> |
| File transfer over HTTP | <input type="checkbox"/> |
| Videoshare | <input type="checkbox"/> |
| De-Register | <input checked="" type="checkbox"/> |
| Regresion | <input type="checkbox"/> |
| New Scenario | <input type="checkbox"/> |

Text Message
Sample IM from Mantica Solutions

Select a local file
Seleccionar archivo Ningún archivo seleccionado

Select a local file
Seleccionar archivo Ningún archivo seleccionado

Select a local video
Seleccionar archivo Ningún archivo seleccionado

Select a local Scenario file
Seleccionar archivo Ningún archivo seleccionado

Apply

Figura 4.3.3.2 Formulario de selección de pruebas.

4.3.3.3 Resultados en tiempo real

Se ha creado una interfaz independiente para visualizar el resultado de las pruebas en formato tabla de resultados.

Cada entrada en la tabla tiene el tiempo en el que ocurrió, una descripción y un estado que puede ser pasado, fallido o la duración si se trata de un KPI.

localhost/PFG_RCS/result: x

localhost/PFG_RCS/results.php

Test Results: A-Party

| Timestamp | Event Description | Status |
|-----------|--|--------|
| 5 | User A | ✓ |
| 5 | RCS Agent initialization started | |
| 5.004 | RCS Agent initialization success | ✓ |
| 5.004 | RCS User initialization | |
| 5.004 | RCS User initialization success | ✓ |
| 5.005 | IMS registration attempt | |
| 5.68 | IMS registration success | ✓ |
| 5.68 | KPI: IMS registration delay | 0.675 |
| 5.68 | RCS Capability Discovery started | |
| 5.68 | RCS Capability Discovery response: -2 [200 OK] | ✓ |
| 5.983 | KPI: RCS Capability Discovery Delay | 0.302 |
| 5.983 | IMS de-registration attempt | |
| 6.633 | IMS de-registration success | ✓ |
| 6.633 | KPI: IMS de-registration delay | 0.651 |
| 6.633 | RCS removing User | |
| 6.633 | RCS User removed success | ✓ |
| 6.634 | RCS Script finish successfully | ✓ |
| 6.635 | RCS creating log file | ✓ |

Figura 4.3.3.3 Tabla de resultados del usuario A

| Timestamp | Event Description | Status |
|-----------|---|--------|
| 0 | User B | ✓ |
| 0 | RCS Agent initialization started | |
| 0.508 | RCS Agent initialization success | ✓ |
| 0.508 | RCS User initialization | |
| 0.508 | RCS User initialization success | ✓ |
| 0.511 | IMS registration attempt | |
| 1.049 | IMS registration success | ✓ |
| 1.049 | KPI: IMS registration delay | 0.538 |
| 1.049 | RCS Capability discovery started | |
| 5.848 | event(10) EVENT_CD_OPTIONS_RX | |
| 5.848 | RCS Capability Discovery request received: RETURN_SUCCESS | ✓ |
| 10.848 | IMS de-registration attempt | |
| 11.545 | IMS de-registration success | ✓ |
| 11.545 | KPI: IMS de-registration delay | 0.696 |
| 11.545 | RCS removing User | |
| 11.545 | RCS User removed success | ✓ |
| 11.545 | RCS Script finish successfully | ✓ |
| 11.546 | RCS creating log file | ✓ |

Figura 4.3.3.4 Tabla de resultados del usuario B

4.3.4 Documentación

Se han realizado dos documentos para detallar el proceso de desarrollo y utilización de la herramienta de monitorización, la memoria del proyecto y el manual del desarrollador.

4.3.4.1 Memoria del proyecto

Contiene toda la información que justifica el desarrollo de la herramienta y la elección de sus componentes y estructura.

4.3.4.2 *Manual del desarrollador*

Contiene toda la información de configuración, desarrollo de código, usuarios y contraseñas de los sistemas y equipos utilizados.

Toda esta información técnica detallada, favorece la continuidad del proyecto por nuevos desarrolladores y ayuda a su comprensión.

5 Despliegue y certificación

5.1 Despliegue

El despliegue definitivo tiene la estructura de red siguiente.

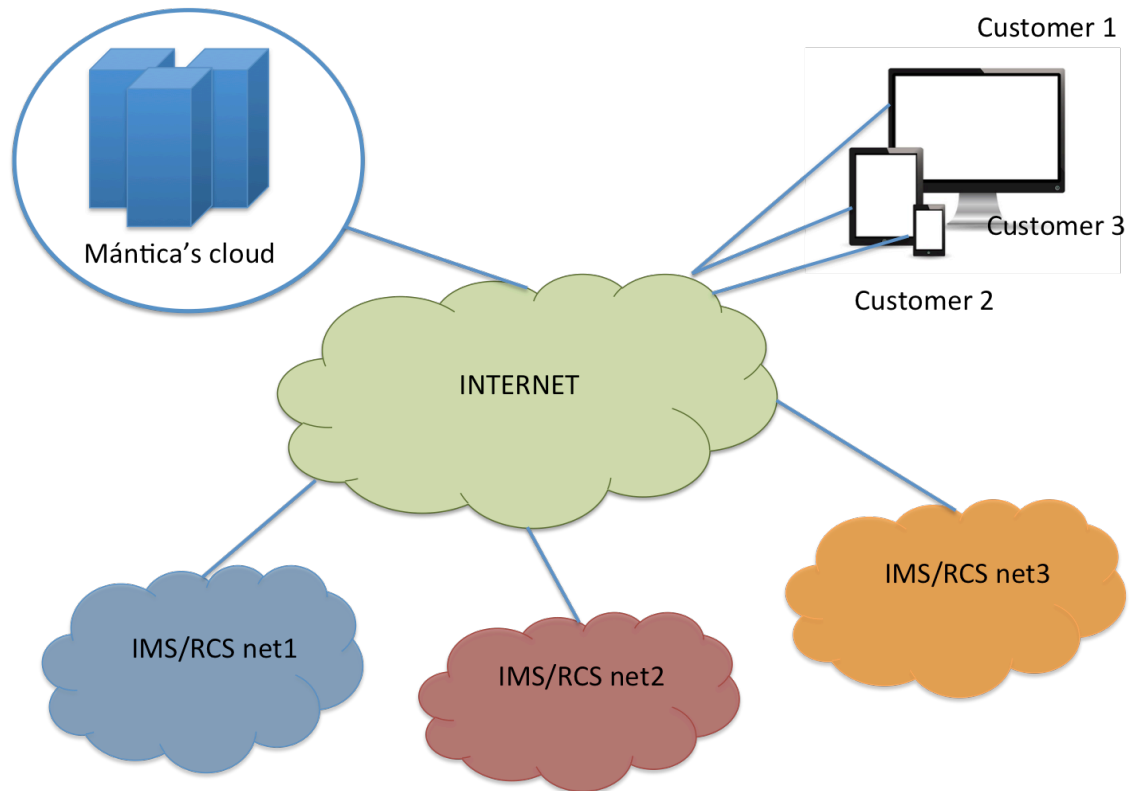


Figura 4.3.4.1 Despliegue de red

Los clientes se conectan a través de una interfaz web en sus equipos locales. Desde allí planifican sus pruebas y envían las instrucciones a servidor de Mántica, desde donde se ejecutan las pruebas. El resultado de la ejecución se envía al navegador del usuario para que la pueda analizar.

5.1.1 Servidor web

Para la implementación del servidor se utiliza un ordenador de sobremesa convencional.

Las características del equipo son:

- PC de sobremesa HP[15] Pavilion 500-260 con procesador Intel® Core™ i5 de 4ª generación.
- Windows 8.1 64
- Intel® Core™ i5-4440
- Gráficos AMD Radeon R5 235 (1 GB ded.)
- Memoria DDR3 de 8 GB
- Disco duro SATA de 1 TB

5.1.2 Conectividad

Al tratarse de un servicio web sólo se requiere conectividad a internet. Se utiliza la conexión de fibra óptica de la empresa.

Además se incluye el dominio mantica-rcs-testing en una herramienta gratuita de DNS dinámico para que sea accesible desde internet.

5.1.3 Usuarios reales de operadores que soporten el servicio

Es necesario adquirir tarjetas SIM[16] para simular usuarios reales en todas las redes en las que se pretende ofrecer el servicio.

Se adquieren dos tarjetas SIM para realizar el desarrollo y dos tarjetas SIM para utilizar en el entorno definitivo.

5.1.4 Sistema de seguridad

Debido a que el sistema es accesible en cualquier momento y desde cualquier lugar se toman varias medidas de seguridad:

- Se establece un único puerto para acceder al equipo desde fuera tanto en el enrutador de la empresa como en el propio equipo.
- Se restringe el acceso a la red local desde el equipo.
- Se crea un proceso que deshabilita el servidor en horario no comercial.
- Se crean alarmas por correo electrónico de alta de nuevos usuarios, acceso de usuarios registrados a la plataforma y accesos al equipo.

5.2 Pruebas

Para garantizar la calidad del servicio ofrecido se implementan varias baterías de pruebas que permiten:

- Certificar que la implementación de la librería cumple con los estándares RCS. Se realizan test regresivos para validar escenarios positivos, la red se comporta de manera esperada, y negativos, existen fallos en la red que provocan flujos de fallo.
- Verificar que cualquier flujo implementado en los casos de prueba desencadena una salida esperada. Se realizan test regresivos para validar salidas positivas y negativas.
- Verificar que la interfaz web es operable con diferentes ordenadores, sistemas operativos, navegadores, versiones de software, etc.
- Verificar el número de clientes que es posible atender con un solo servidor.

6 Planificación y presupuesto

6.1 Planificación

Se estimó una duración de 5 meses para la realización del proyecto.

| Tarea | Tiempo (semanas) |
|--|------------------|
| Estudio de sistemas alternativos | 2 |
| Recopilación de datos | 1,0 |
| análisis comparativo | 1,0 |
| Análisis de requisitos | 2 |
| Definición de requisitos de la herramienta | 1,0 |
| Estudio de herramientas de desarrollo | 0,5 |
| Elección de herramientas de desarrollo | 0,5 |
| Desarrollo | 8 |
| Casos de prueba | 4,0 |
| Resultados | 2,0 |
| Interfaz web | 2,0 |
| Pruebas de validación y corrección de errores | 3 |
| Desarrollo de las pruebas | 1,5 |
| Ejecución | 0,5 |
| Corrección de errores | 1,0 |
| Despliegue | 1 |
| Instalación y configuración | 0,5 |
| Comprobaciones | 0,5 |
| Pruebas de certificación y corrección de errores | 3 |
| Desarrollo de las pruebas | 1,5 |
| Ejecución | 0,5 |
| Corrección de errores | 1,0 |
| Imprevistos | 1 |
| Imprevistos | 1,0 |
| Total | 20 |

Figura 5.1.4.1 Tabla de planificación

6.2 Presupuesto

Para la realización de un proyecto de desarrollo tenemos que tener en cuenta los costes de 3 tipos. Costes hardware, costes software y costes de personal.

6.2.1 Costes de hardware

Aplicaremos la siguiente fórmula para calcular los costes, fijando un periodo de amortización de 5 años y una utilización dedicada de los equipos durante el proyecto (5 meses).

$$\text{Coste} = \left(\frac{\text{Utilización del equipo}}{\text{Periodo de depreciación}} \right) * \text{Coste del equipo (sin IVA)}$$

| Hardware | Coste total sin IVA (€) | Utilización (meses) | Periodo de depreciación (meses) | Coste (€) |
|---------------------------|-------------------------|---------------------|---------------------------------|----------------|
| Desarrollo | | | | |
| Ordenador portátil ASUS | 419,30 € | 5 | 60 | 34,94 € |
| Monitor LCD | 140,00 € | 5 | 60 | 11,67 € |
| Ratón y Teclado | 28,00 € | 5 | 60 | 2,33 € |
| SIMs | 14,00 € | 5 | 60 | 1,17 € |
| Despliegue | | | | |
| Ordenador de sobremesa HP | 419,30 € | 5 | 60 | 34,94 € |
| SIMs | 14,00 € | 5 | 60 | 1,17 € |
| Total | | | | 86,22 € |

Figura 6.2.1.1 Tabla de costes Hardware

6.2.2 Coste de software

No hay costes software asociados, ya que todas las herramientas que hemos utilizado son de software libre. Además la utilización de alguna herramienta licenciada que posea la empresa la trataremos en la totalización.

6.2.3 Costes de personal

Para los costes de personal se incluye la labor del jefe de proyectos en tareas de gestión y se dividen las tareas del alumnos durante la beca. Aunque calculamos un precio medio por hora en base al salario:

$$\text{Retribución mensual (500€)} = \frac{4 \text{ Horas}}{\text{Día}} \times \frac{5 \text{ Días}}{\text{Semana}} \times 4 \text{ Semanas}$$

| Función | €/hora | Horas | Total |
|--------------------------------|---------|-------|------------|
| Jefe del proyecto | | | |
| Ingeniero gestor de proyectos | 60,00 € | 40 | 2.400,00 € |
| Becario | | | |
| Ingeniero analista y diseñador | 6,25 € | 80 | 500,00 € |
| Ingeniero de desarrollo | 6,25 € | 200 | 1.250,00 € |
| Ingeniero de pruebas | 6,25 € | 120 | 750,00 € |
| Total | | | 4.900,00 € |

Figura 6.2.3.1 Tabla de costes de personal

6.2.4 Totalización de costes

Para calcular el coste total del proyecto se tiene en cuenta un sobrecoste derivado de la utilización de material de la empresa, además del riesgo asociado a cualquier proyecto y un 21% de impuestos.

| Concepto | Coste total |
|---------------------------------|-------------------|
| Coste Hardware | 86,22 € |
| Coste Software | 0,00 € |
| Coste Personal | 4.900,00 € |
| Otros costes (10% del personal) | 490,00 € |
| Subtotal 1 | 5.476,22 € |
| Beneficio de la empresa(20%) | 1.095,24 € |
| Subtotal 2 | 6.571,46 € |
| Cuantificación del riesgos(10%) | 657,15 € |
| Base imponible | 7.228,61 € |
| IVA (21%) | 1.518,01 € |
| Total | 8.746,62 € |

Figura 6.2.4.1 Tabla de costes totales

7 Futuro y Conclusiones

7.1 Futuro

Los clientes están muy interesados en este tipo de herramientas y escuchando sus necesidades podemos extraer ciertas líneas de seguimiento para mejorar la herramienta.

7.1.1 Mejoras hardware

- Incorporar pruebas usando acceso de datos
 - Para ello habría que incorporar módems para realizar pruebas en 2G,3G y 4G.
- Incorporar un sistema de gestión de tarjetas SIM.
 - Si el crecimiento de los clientes es grande y se ofrecen pruebas de interoperabilidad la demanda de tarjetas SIM se multiplica. Es inviable gestionar el sistema manualmente o proporcionar un modem por SIM.
 -

7.1.2 Mejoras software

- Incorporar pruebas de interoperabilidad con otras redes en todo el mundo.
 - Ya que el sistema está destinado a ser interoperable, deberíamos proporcionar un portfolio de redes acreditadas contra las que probar interoperabilidad.
 -
- Incorporar nuevos escenarios de prueba.
 - Una parte importante de la actividad del departamento es realizar pruebas de certificación de clientes y redes. Se podría automatizar una batería de pruebas que cumpla las especificaciones del organismo certificador
 -
- Mejorar el sistema de auto generación de escenarios.
 - Hacer un sistema de gestión de escenarios más amplio y completo.
 -
- Incorporar un planificador de pruebas.
 - Los clientes también demandan la monitorización continua. Para ello se podría incorporar una herramienta software que planifique el momento del lanzamiento de cada prueba sin que afecte a las que se están ejecutando y de manera autónoma.
 -
- Incorporar diagnósticos avanzados
 - Proveer una información avanzada más completa de la causa del fallo.
 - Determinar a que nivel está afectando un fallo (TCP, IP,SIP, MSRP).

- Generar alarmas que alerten de fallos sistemáticos e indisponibilidades para realizar intervenciones antes de que los clientes se vean afectados por los fallos de servicio.
- Incorporar un Sistema de estadísticas más moderno
 - Incluir histórico de todas las pruebas realizadas y realizar agrupaciones y gráficas que ayuden a diagnosticar el servicio de manera genérica sin tener que entrar a los detalles técnicos.
 - Identificar los fallos más comunes para eliminarlos.

7.1.3 Nuevas líneas de negocio

- Ampliación a otros servicios con bases similares, por ejemplo VoLTE[17]

7.2 Conclusiones

El propósito principal del proyecto, realizar una herramienta de monitorización activa de servicios RCS, se ha cumplido en requisitos y tiempos.

Además los clientes muestran interés en su uso y evolución, lo que supone un incentivo para seguir trabajando en esta línea.

Por otro lado, al tratarse de servicios avanzados de red, la complejidad es muy alta y la demanda de sistemas de altas prestaciones que sean capaces de realizar muchas pruebas y recopilar toda la información que se genera a su alrededor es imprescindible. Los siguientes pasos serán clave para determinar si la compañía se posiciona con un buen producto y rentabiliza el proceso de investigación y desarrollo o no.

8 Bibliografía

3GPP. (2 de 09 de 2014). *IMS*. Obtenido de <http://www.3gpp.org/technologies/keywords-acronyms/109-ims>. [4]

3GPP. (s.f.). *SIM*. Obtenido de Wikipedia: http://es.wikipedia.org/wiki/Tarjeta_SIM [16]

Association, GSMA. (s.f.). *Rich Communication Services*. Recuperado el 10 de 09 de 2014, de GSMA: <http://www.gsma.com/network2020/rcs/> [1]

Bourdon, R. (12 de 09 de 2014). Obtenido de Wampserver: <http://www.wampserver.com/> [7]

GASMA Associoation. (s.f.). *volte*. Obtenido de GSMA: <http://www.gsma.com/network2020/volte/> [17]

GSMA Association. (5 de 09 de 2014). Obtenido de Joyn: <http://www.joynus.com/es/> [5]

HP. (s.f.). *HP*. Recuperado el 05 de 09 de 2014, de HP: www.hp.com [15]

IETF. (s.f.). *RTP*. Recuperado el 10 de 09 de 2014, de ietf: <https://www.ietf.org/rfc/rfc3550.txt> [13]

IETF. (s.f.). *tools*. Recuperado el 10 de 09 de 2014, de MSRP: <http://tools.ietf.org/html/rfc4975> [12]

Microsoft. (s.f.). *Windows*. Recuperado el 20 de 09 de 2014, de Microsoft: <http://windows.microsoft.com/es-es/windows/home> [6]

Nagios Enterprises, LLC. (s.f.). *Nagios*. Recuperado el 18 de 09 de 2014, de Nagios: <http://www.nagios.org/> [2]

Reh, F. J. (s.f.). *keyperfindic*. Recuperado el 08 de 09 de 2014, de management.about:
<http://management.about.com/cs/generalmanagement/a/keyperfindic.htm> [14]

Tibbo Technology Inc. (6 de 09 de 2014). *Aggregate network monitoring*. Obtenido de
http://aggregate.tibbo.com/solutions/network_management/network_monitoring.html [3]

w3schools. (s.f.). *w3schools*. Recuperado el 09 de 09 de 2014, de HTML5:
http://www.w3schools.com/html/html5_intro.asp [8]

w3schools. (s.f.). *w3schools*. Obtenido de CSS3:
http://www.w3schools.com/css/css3_intro.asp [9]

w3schools. (s.f.). *w3schools*. Recuperado el 09 de 09 de 2014, de JQuery:
<http://www.w3schools.com/jquery/default.asp> [11]

w3schools. (s.f.). *www.w3schools*. Recuperado el 09 de 09 de 2014, de JS:
<http://www.w3schools.com/js/default.asp> [10]

